文章编号：2096-1618(2020)02-0158-05

基于关键词抽取的网络博客自动文摘算法的研究

李 敏， 陶宏才
(西南交通大学信息科学与技术学院，四川 成都 611756)

摘要：TextRank算法基于图论，考虑文本的整体结构，而关键词与文本主题紧密关联。网络博客作为一种新兴的出版方式，与新闻、专业论文等文本不同，其编辑方式更为随意，没有传统意义上的一般格式。将关键词抽取与TextRank算法结合起来，提出一种适用于博客文本的基于关键词抽取的自动文摘算法。首先通过TextRank算法抽取文本关键词，用BM25算法计算句子相似度。然后，以句子相似度为权重构建带权图，迭代计算获取TextRank评分。将TextRank评分与关键词评分相加得到句子最终得分，选择评分最高的前 i 个句子，按照句子在原文中的顺序输出得到自动文摘。通过ROUGE工具的测评，设计对比实验证明算法效果良好。

关 键 词：自动文摘；TextRank；关键词；BM25；ROUGE
中图分类号：TP391.1 **文献标志码**：A
doi：10.16836/j.cnki.jcuit.2020.02.006

0 引言

随着互联网时代的来临，网络已经成为了人们获取信息的重要途径。网络快速发展带来信息量的剧增，用户在海量数据中如何精准地获取所需的信息，俨然已经成为了一个值得研究与关注的问题。设想一下，人们在正式阅读文本之前，如果能够提前阅读原文本的文摘，那么就能够在一个较短的时间内掌握文章的主要内容，判断该文本对自身的阅读价值，提高阅读的效率。自动文摘就是指通过特定算法利用计算机强大的计算功能生成文本摘要的过程[1]。

博客(Blog)指网络日志，是一种传播个人思想，带有知识集合链接的出版方式[2]。与新闻以及专业论文等文本不同，博客的编辑方式更为随意，没有传统意义上的一般格式。因此，针对于这类文本的自动文摘技术的研究，本文没有选择篇章结构等文本特征作为影响因素，而是将关键词这一文本普适特征与自动文摘技术结合起来进行研究。自动文摘技术分为生成式与抽取式，本文是针对基于关键词的抽取式自动文摘技术的研究。

1 相关工作与理论

1.1 国内外研究现状

自动文摘技术的发展可以追溯到计算机诞生之时。根据所摘取的原始文本的数量，可以将其分为单文本摘要[3]与多文本摘要[4]。根据文摘的生成方式又可以划分为抽取式文摘[5]和生成式文摘。

1958年由H. P. Luhn[6]发表的《The Automatic Creation of Literary Abstracts》揭开了计算机自动文摘技术研究的序幕。H. P. Edmundson[7]基于Lunh的思想，首次考虑文本本身特征，提出了结合文本特征的词频统计自动文摘技术。随着机器学习的发展，Kupiec与Radev首次将机器学习技术运用于自动文摘生成领域[8]。

2004年，Mihalcea R等[9]在PageRank[10]算法的基础上，提出了著名的TextRank算法。TextRank算法将文本片段作为图的节点，以节点之间的相似度作为边的权重。通过算法迭代计算，得到节点权重排序结果并输出。曹阳[11]提出关于相似度的计算，主要有基于信息量的相似度的计算、基于编辑距离的相似度的计算、基于语义词典的相似度的计算、基于BM25的相似度的计算等四种方法，其中基于BM25的计算方法具有良好的表现。

1.2 TextRank算法

TextRank算法作为一种无监督学习方法，其思想基于Google创始人为了解决搜索引擎对网页的排序而提出的PageRank算法。

对于一个网页而言，可以猜想当其他网页对它的引用越多那么该网页就越重要，也就是 *PR* 值越高。图1为三个网页的跳转情况，其中a网页指向b、c网页，d、c网页又指向a网页。

收稿日期：2019-12-26
基金项目：国家自然科学基金资助项目(61806170)

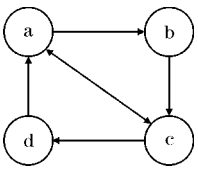

图 1　网页跳转图

网页 *PR* 值计算公式为

$$PA(a)_{i+1}=\sum_{i=0}^{n}\frac{PR(T_i)}{L(T_i)} \tag{1}$$

其中,$PR(T_i)$ 为指向 a 节点的其他节点的 *PR* 值,$L(T_i)$ 为节点的出链数。一般而言,初始 *PR* 值为 $1/N$,N 为总节点数。以 a 节点为例,*PR* 值计算过程如下:

$i=0$,　$PR=1/4$

$i=1$,　$PR=\frac{PR(c)_0}{L(c)}+\frac{PR(d)_0}{L(d)}$

$=\frac{1}{4}\div 2+\frac{1}{4}\div 1=\frac{3}{8}$

…

经过数次迭代计算后,每个网页都会得到一个稳定的 *PR* 值,通过对 *PR* 值的排序可以得到网页的重要性排序。将图 1 中的网页换作文本中的句子,利用句子之间的相似度作为边的权重,那么经过迭代计算后,将最终的边权重计算结果作为对应句子节点的权重,权重排序靠前的句子肯定与文中其他句子相似度高、关系更为紧密。TextRank 算法按式(2)通过抽取排名靠前的句子,就能够得到文本文摘。

$$WS(V_i)=(1-d)+d\cdot\sum_{V_j\in In(V_i)}\frac{w_{ji}}{\sum_{V_k\in Out(V_j)}w_{jk}}WS(V_j) \tag{2}$$

其中,$In(Vi)$ 可以理解为指向该句子节点的其它句子节点,如图 1 中指向 a 网页的 c、d 节点;而 $Out(V_j)$ 则为 c、d 节点所指向的节点。d 为修正参数,是对 Dead ends 与 Spider Traps 问题的处理,d 一般在[0.8,0.9]取值,文中取0.85。

传统 TextRank 算法使用式(3)计算句子相似度:

$$Similarity(S_i,S_j)=\frac{|\{t_k\vee t_k\in S_i\wedge t_k\in S_j\}|}{\lg(|S_i|)+\lg(|S_j|)} \tag{3}$$

1.3　BM25 算法

BM25[12]算法常用于信息检索领域,用来对相似度进行打分。计算一个句子中所有的词与文档的相关度后累加,如式(4)所示。

$$Score(Q,d)=\sum_{i}^{n}W_i\cdot R(q_i,d) \tag{4}$$

其中 W_i 是词语的权重,一般为逆向文档频率 IDF(Inverse Document Frequency)值,由式(5)计算得到。

$$IDF(q_i)=\lg\frac{N+0.5}{n(q_i)+0.5} \tag{5}$$

其中,N 为总文档数,$n(q_i)$ 为包含该词语的文档数。为避免分母为 0 的错误,对分子分母同时加上0.5进行调节。由式(5)可知,当总的文档数越大,而包含该词的文档数量越少时,*IDF* 值就越大。比如像“的、地、得”等这一类词在很多文档中都存在,即$n(q_i)$值越大,那么对应的 *IDF* 值就越小,对指定文档的重要性就越小。

$$R(q_i,d)=\frac{f_i\cdot(k_1+1)}{f_i+K}\cdot\frac{qf_i\cdot(k_2+1)}{qf_i+k_2} \tag{6}$$

$$K=k_1\cdot(1-b+b\cdot\frac{dl}{\text{avg}(dl)}) \tag{7}$$

其中 f_i 为词语在文档 d 中出现的频率,qf_i 为词语在查询语句中出现的频率,式(7)中 dl 为文档 d 的长度,avg(dl)为所有文档的平均长度。

$$Score(Q,d)=\sum_{i}^{n}IDF(q_i)\cdot\frac{f_i\cdot(k_1+1)}{f_i+k_1\cdot(1-b+b\cdot\frac{dl}{\text{avg}(dl)})} \tag{8}$$

其中 k_1、k_2、b 为调节参数,一般 $k_1=1$,$k_2=1$,$b=0.75$。

2　一种基于关键词的文本摘要新算法

2.1　问题的提出

TextRank 的优点在于能够覆盖文本全局结构且不需要其他先验信息,不需要经过训练,适用于各类文本。同样,无论什么类型的文本,关键词都能够十分直观地表现文本的研究内容。因此,可以假设,将关键词结合 TextRank 算法实现自动文摘能够提升算法表现。

2.2　关键词的抽取

与自动文摘算法一样,文中关键词的抽取也利用 TextRank 算法实现,不同的地方在于当以词语作为节点在构造图的时候,图的边就变成了词语与词语之间的关系,并不能够简单将词语的相似性作为权重,因此这里采用窗口投票的方式实现[13]。算法步骤如下:

步骤 1　将文本分割为句子,其中使用的分隔符可以自行设定,选取″[,,。::“”??!!;;]″等符号,对句子进行分词并去停用词。得到句子 S_i 的功能词集合 $\{w_1,w_2,\cdots,w_n\}$。

步骤 2　建立大小为 k 的窗口,每个单词对其前后 k 个单词进行投票。通过投票结果以步骤 1 得到的所有词为节点构造图,两个节点之间如果存在投票关系,则建立一条边。

步骤 3　对步骤 2 得到的图,根据式(1)进行迭代

计算。得到最终的排序结果后,将排名前 i 的词语作为关键词输出。

2.3 新算法整体处理流程描述

算法流程图如图 2 所示。

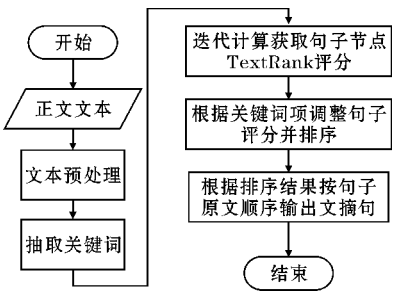

图 2 算法流程图

步骤 1 文本预处理。对每一个句子进行分词、去停用词,得到句子集合 $S=\{S_1, S_2,\cdots,S_n\}$,其中 S_i 为词语组成的数组。

步骤 2 抽取关键词。通过2.2节方法抽取文本关键词得到结果集 $KW=\{kw_1, kw_2,\cdots,kw_k\}$。

步骤 3 迭代计算获取句子节点 TextRank 评分。运用式(8)计算候选句子集的 BM25 相似度矩阵,示例如图 3 所示。可以用矩阵来表示句子节点之间由相似度所构造的图,图 3 的 BM25 矩阵 $\boldsymbol{A}$ 中,a_{ij}表示 S_i 与 S_j 之间的 BM25 值,该值的大小表示 S_i 与 S_j 之间的相似度,即构造图边的权重。当 $a_{ij}=0$ 时则表示在这两个节点之间没有边,矩阵的运用使得存储计算更加方便。然后将句子评分数组 Vertex 初始化为全 1。

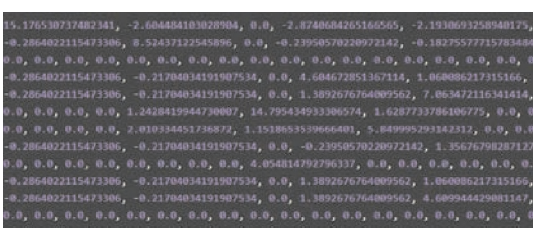

图 3 BM25 矩阵 $\boldsymbol{A}$ 示例

根据式(2)对句子节点评分进行迭代计算直至结果稳定,结果稳定的条件一般为迭代计算过程中每个节点的计算误差小于给定误差值。

步骤 4 根据关键词项调整句子评分并排序。将步骤 2 与步骤 3所得结果带入式(9)计算句子节点的最终权重 $Weight=\{ws_1, ws_2,\cdots, ws_n\}$。

$$Weight(S_i)=t\cdot Vertex(S_i)+(1-t)\cdot NK(S_i) \quad (9)$$

其中,$Vertex(S_i)$为步骤 3 通过 TextRank 算法所得到的句子评分,$NK(S_i)$为 S_i 中所包含关键词的数量。t 为调节参数,根据后续实验结果一般取 $t=0.85$。

步骤 5 对步骤 4 结果集 *Weight* 按照句子在原文中的顺序输出前 i 个句子作为文摘句。

3 实验及分析

3.1 Rouge 评价方法

自动文摘的评价方法主要分为内部评价方法与外部评价方法。内部评价方法直接分析文摘本身的质量,而外部评价方法则将生成的文摘运用到某一任务中,如文本分类,通过评价任务的完成结果来评价文摘[14]。常用的内部评价方法包括 ROUGE、Edmundson,文中采用 ROUGE 评价方法。

ROUGE 基于摘要中 n 元词(n-gram)的共现信息评价文摘,是一种面向 n 元词召回率的评价方法[15],包括 ROUGE-1、ROUGE-2、ROUGE-3、ROUGE-L 等一系列评价方法,计算公式为

$$ROUGE\text{-}N=\frac{\sum_{S\in\{\text{RefSummaries}\}}\sum_{n\text{-gram}\in S}Countmatch(n\text{-}gram)}{\sum_{S\in\{\text{RefSummaries}\}}\sum_{n\text{-gram}\in S}Count(n\text{-}gram)} \quad (10)$$

其中,*n-gram* 表示 n 元词,{Ref Summaries}为参考摘要,即事先获得的标准摘要,*Countmatch*(*n-gram*)表示系统摘要和参考摘要中同时出现 *n-gram* 的个数,*Count*(*n-gram*)则表示参考摘要中出现的 *n-gram* 个数[15]。

选择 ROUGE-1、ROUGE-2、ROUGE-3、ROUGE-4、ROUGE-L 作为评价指标,使用基于 ROUGE-1.5.5 python 实现的 pyrouge 开源工具。

3.2 算法相关参数实验

3.2.1 关键词抽取数量实验

为评价关键词数目对实验结果的影响,分别测试关键词数目 n 为 3、4、5 时的实验表现。实验中抽取的文摘句数量都为文本长度的 10%,取式(9)中 $t=0.75$。实验数据为 NLPCC2015 发布的公开数据集,其中包括 140 篇新闻文本及给定文摘,另从新浪博客爬取娱乐、文史、教育、体育 4 个领域的 200 篇文章用于实验。实验结果如表 1 所示, 表中 $R_1, R_2\cdots, R_L$ 分别表示 ROUGE-1、ROUGE-2,…,ROUGE-L。

表 1 关键词数量实验结果

n	R_1	R_2	R_3	R_4	R_L
3	0.5151	0.3174	0.2158	0.1641	0.4855
4	0.5353	0.3464	0.2448	0.1936	0.5051
5	0.5167	0.3388	0.2412	0.1930	0.4843

由表 1 可知,当关键词数量 n 为 4 时算法的效果最好。

3.2.2　评分参数 t 实验

为了评价式(9)中 t 值对实验结果的影响。分别测试 t 为0.65、0.75、0.85时的算法效果。实验中关键词数量 $n=4$,文摘句占比 10%,实验结果如表 2 所示。

表 2　权重参数 t 实验结果

t	R_1	R_2	R_3	R_4	R_L
0.65	0.5084	0.3268	0.2276	0.1796	0.4753
0.75	0.5353	0.3464	0.2448	0.1936	0.5051
0.85	0.5393	0.3478	0.2438	0.1914	0.5058

由表 2 可知,随着 t 的增加实验表现越来越好,因此增设 $t=0.9$进行对比,实验结果如表 3 所示。

表 3　权重参数 t 实验结果

t	R_1	R_2	R_3	R_4	R_L
0.9	0.5311	0.3468	0.2434	0.1914	0.4975

根据增设实验可知,当 t 取0.85时算法表现最好。

3.3　算法对比实验及结果分析

选择经典 TextRank 算法、基于 BM25 改进 TextRank 算法[11]和基于关键词过滤非相关句算法(此算法是为做对比实验而专门设计的)与文中算法进行对比实验。其中,专门设计的基于关键词过滤非相关句算法的主要思想是将抽取的关键词用于在运行 TextRank 算法之前,过滤掉文本非相关句得到候选句子集,再对候选句集进行 TextRank 算法的迭代计算。设计该对比算法的目的,是为考查在保证算法准确性表现较优的情况下能否降低时间复杂度,优化算法性能。

基于关键词过滤非相关句算法和文中算法根据关键词在 TextRank 算法的使用前后顺序,分别命名为 KWBefTextRank、KWAftTextRank。经典 TextRank 算法、基于 BM25 改进 TextRank 算法则分别命名为 TextRank、BM25。其中,实验数据集与参数实验保持一致,KWAftTextRank 参数基于3.2节实验结果分别选择关键词数量 $n=4$、$t=0.85$。

4 种算法的对比实验结果如表 4 所示。由表 4 可知,文中算法通过 ROUGE 工具测评得到的各项 ROUGE 指标,均优于其他 3 个对比算法,表现良好。

由于 KWBefTextRank 算法的表现甚至差于传统 TextRank 算法,因此不再考虑其时间复杂度上可能存在的优势。

表 4　4 种算法对比实验结果

算法	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L
TextRank	0.51800	0.31181	0.23260	0.18226	0.48381
BM25	0.52311	0.30274	0.22017	0.16803	0.48907
KWBefTextRank	0.51727	0.32532	0.22198	0.16188	0.48097
KWAftTextRank	0.53930	0.34775	0.24383	0.19139	0.50582

4　结束语

选择基于图的 TextRank 算法以及与文本主题密切相关的关键词,从文本的整体结构脉络出发,不需要其他先验信息,不需进行大量计算,针对无明显文本篇章结构特征的网络博客文本,提出了一种基于关键词抽取的自动文摘算法。通过对比试验以及 ROUGE 测评,相较于传统 TextRank 算法、改进 BM25 算法以及对比算法 KWBefTextRank,文中算法的准确性更高。

当然,文中算法也存在一些不足。例如,算法实验结果可能比较依赖于关键词抽取的准确性,而关键词的抽取又依赖于文本预处理结果。因此,后续工作可以考虑提升文本预处理的效果以及关键词抽取算法的效果,以进一步提升本文算法的表现。

参考文献:

[1]　刘家益,邹益民. 近 70 年文本自动摘要研究综述[J]. 情报科学,2017,35(7):154-161.

[2]　百度百科. 博客[DB/OL]. https://baike.baidu.com/item/%E5%8D%9A%E5%AE%A2/124?fr=aladdin,2019-12-20.

[3]　刘海燕,张钰. 基于 LexRank 的中文单文档摘要方法[J]. 兵器装备工程学报,2017,38(6):85-89.

[4]　付玲,张晖. 结合 LDA 和谱聚类的多文档摘要[J]. 计算机工程与应用,2013,49(16):142-145.

[5]　郑义. 多媒体信息自动摘要及其相关技术研究[D]. 上海:复旦大学,2003.

[6] Luhn H P. The automatic creation of literature abstract[J]. IBM Journal of Research and Development,1958,2(2):159-165.
[7] EDMUNDSON H P. New Methods in Automatic Extracting[J]. 1969,16(2):264-285.
[8] 张静静. 基于知网文本相似度的文摘自动评测方法研究[D]. 北京:中国石油大学,2011.
[9] Mihalcea R,Rada,Tarau,Paul. TextRank:Bringing Order into Texts[J]. Unt Scholarly Works,2004:404-411.
[10] PAGE, L. The PageRank Citaition Ranking: Bringing Order to the Web, Online manuscript [J]. Stanford Digital Libraries Working Paper. 1998,9(1):1-14.
[11] 曹洋. 基于TextRank算法的单文档自动文摘研究[D]. 南京:南京大学,2016.
[12] Robertson S E,Walker S. Beaulieu M,et al. Okapi at TREC5[J]. 1996.
[13] CSDN. TextRank 关键词提取算法[DB/OL]. https://blog. csdn. net/qq_34333481/article/details/85705039,2019-12-21.
[14] 张瑾,王小磊,许洪波. 自动文摘评价方法综述[J]. 中文信息学报,2008(3):81-88.
[15] CSDN. 自动文档摘要评价方法:Edmundson,ROUGE[DB/OL]. https://blog. csdn. net/weixin_33712987/article/details/93248861,2019-12-21.

Research on Automatic Digest Algorithm of Web Blog based on Keyword Extraction

LI Min, TAO Hongcai
(School of Information Science & Technology, Southwest Jiaotong University, Chengdu 611756, China)

Abstract: The TextRank algorithm is based on graph theory, considering the overall structure of the text. Keywords are closely related to the text theme. As a new publishing method, online blogs are different from texts such as news and papers, and their editing methods are more casual. There is no general format in the traditional sense. This paper combines keyword extraction with TextRank algorithm, and proposes an automatic abstracting algorithm based on keyword extraction suitable for blog text. First, keywords are extracted through the TextRank algorithm, and sentence similarity is calculated using the BM25 algorithm. The sentence similarity is used to construct weighted graphs for weights, and iterative calculations are used to obtain TextRank scores. The TextRank score and the keyword score are added to the final sentence score, and the top i sentences with the highest score are selected and output in the order of the original text to obtain automatic digests. Through the evaluation of ROUGE tools, comparison experiments show that the algorithm works well.
Keywords: automatic digest; TextRank; keywords; BM25; ROUGE